

Compte rendu du Tp Bataille navale

Sommaire

Introduction

Dans ce TP, j'ai développé un jeu de Bataille Navale en Java pour me familiariser avec la programmation orientée objet, la gestion des tableaux multidimensionnels et les structures de contrôle.

Etape 1

J'ai commencé par créer deux grilles de jeu 5x5 - une pour le joueur et une pour l'ordinateur :

```
int[][] grilleJoueur = new int[5][5];
int[][] grilleOrdinateur = new int[5][5];
```

J'ai ensuite initialisé toutes les cases à zéro en utilisant des boucles imbriquées :

```
for (int i = 0; i < 5; i++) {
    for (int j = 0; j < 5; j++) {
        grilleJoueur[i][j] = 0;
        grilleOrdinateur[i][j] = 0;
    }
}
```

Etape 2

J'ai implémenté un système permettant au joueur de placer ses 5 bateaux :

```
int bateauxPlaces = 0;
Scanner scanner = new Scanner(System.in);

while (bateauxPlaces < 5) {
    System.out.println("Placement du bateau " + (bateauxPlaces + 1));

    // Saisie des coordonnées
    System.out.print("Ligne (0-4) : ");
    int ligne = scanner.nextInt();
    System.out.print("Colonne (0-4) : ");
    int colonne = scanner.nextInt();

    // Validation des coordonnées
    if (ligne < 0 || ligne > 4 || colonne < 0 || colonne > 4) {
        System.out.println("Coordonnées invalides !");
        continue;
    }

    // Vérification case libre
    if (grilleJoueur[ligne][colonne] == 1) {
        System.out.println("Case déjà occupée !");
        continue;
    }

    // Placement du bateau
    grilleJoueur[ligne][colonne] = 1;
    bateauxPlaces++;
    System.out.println("Bateau placé avec succès !");
}
```

Etape 3

J'ai programmé le placement automatique des bateaux de l'ordinateur :

```
int bateauxOrdinateur = 0;
Random random = new Random();

while (bateauxOrdinateur < 5) {
    int ligne = random.nextInt(5);
    int colonne = random.nextInt(5);

    if (grilleOrdinateur[ligne][colonne] == 0) {
        grilleOrdinateur[ligne][colonne] = 1;
        bateauxOrdinateur++;
    }
}
```

Etape 4

J'ai créé une fonction pour afficher la grille du joueur :

```
public static void afficherGrille(int[][] grille) {
    System.out.print("  ");
    for (int i = 0; i < 5; i++) {
        System.out.print(i + " ");
    }
    System.out.println();

    for (int i = 0; i < 5; i++) {
        System.out.print(i + " ");
        for (int j = 0; j < 5; j++) {
            if (grille[i][j] == 1) {
                System.out.print("B ");
            } else {
                System.out.print("~ ");
            }
        }
        System.out.println();
    }
}
```

Tableau du joueur :
0 1 2 3 4
0 ~ ~ ~ ~
1 0 ~ ~ 0 ~
2 ~ ~ ~ 0 ~
3 ~ ~ ~ 0 ~
4 ~ 0 ~ ~ ~

Etape 5

Pour la grille de l'ordinateur, j'ai créé une version cachée :

```
public static void afficherGrilleCachee(int[][] grille) {
    System.out.print("  ");
    for (int i = 0; i < 5; i++) {
        System.out.print(i + " ");
    }
    System.out.println();

    for (int i = 0; i < 5; i++) {
        System.out.print(i + " ");
        for (int j = 0; j < 5; j++) {
            if (grille[i][j] == 2) {
                System.out.print("X ");
            } else if (grille[i][j] == 3) {
                System.out.print("O ");
            } else {
                System.out.print("? ");
            }
        }
        System.out.println();
    }
}
```

Etape 6

J'ai implémenté le mécanisme de tir du joueur :

```
public static boolean tourJoueur(int[][] grilleOrdinateur, Scanner scanner) {
    System.out.println("\n--- VOTRE TOUR ---");
    afficherGrilleCachee(grilleOrdinateur);

    int ligne, colonne;
    do {
        System.out.print("Ligne à attaquer (0-4) : ");
        ligne = scanner.nextInt();
        System.out.print("Colonne à attaquer (0-4) : ");
        colonne = scanner.nextInt();
    } while (ligne < 0 || ligne > 4 || colonne < 0 || colonne > 4);

    // Simulation du tir
    System.out.println("Tir en cours...");
    try {
        Thread.sleep(2000);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }

    // Résultat du tir
    if (grilleOrdinateur[ligne][colonne] == 1) {
        System.out.println("TOUCHÉ !");
        grilleOrdinateur[ligne][colonne] = 2;
        return true;
    } else {
        System.out.println("MANQUÉ !");
        if (grilleOrdinateur[ligne][colonne] == 0) {
            grilleOrdinateur[ligne][colonne] = 3;
        }
        return false;
    }
}
```

Etape 7

J'ai programmé l'intelligence artificielle de l'ordinateur :

```
public static boolean tourOrdinateur(int[][] grilleJoueur) {  
    System.out.println("\n--- TOUR DE L'ORDINATEUR ---");  
    System.out.println("L'ordinateur réfléchit...");  
  
    try {  
        Thread.sleep(2000);  
    } catch (InterruptedException e) {  
        e.printStackTrace();  
    }  
  
    Random random = new Random();  
    int ligne, colonne;  
  
    do {  
        ligne = random.nextInt(5);  
        colonne = random.nextInt(5);  
    } while (grilleJoueur[ligne][colonne] == 2 || grilleJoueur[ligne][colonne] == 3);  
  
    System.out.println("L'ordinateur tire en (" + ligne + "," + colonne + ")");  
  
    if (grilleJoueur[ligne][colonne] == 1) {  
        System.out.println("TOUCHÉ !");  
        grilleJoueur[ligne][colonne] = 2;  
        return true;  
    } else {  
        System.out.println("MANQUÉ !");  
        grilleJoueur[ligne][colonne] = 3;  
        return false;  
    }  
}
```

Etape 8

J'ai finalisé le jeu avec la boucle principale et l'option de rejouer :

```
public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    boolean rejouer;

    do {
        rejouer = false;

        // Initialisation
        int[][] grilleJoueur = initialiserGrille();
        int[][] grilleOrdinateur = initialiserGrille();

        placementBateauxJoueur(grilleJoueur, scanner);
        placementBateauxOrdinateur(grilleOrdinateur);

        // Déroulement de la partie
        int scoreJoueur = 0, scoreOrdinateur = 0;

        while (scoreJoueur < 5 && scoreOrdinateur < 5) {
            if (tourJoueur(grilleOrdinateur, scanner)) scoreJoueur++;
            if (scoreJoueur == 5) break;
            if (tourOrdinateur(grilleJoueur)) scoreOrdinateur++;
        }

        // Résultat
        System.out.println(scoreJoueur == 5 ?
            "\nFÉLICITATIONS ! Vous avez gagné !" :
            "\nDÉFAITE ! L'ordinateur a gagné !");

        // Rejouer ?
        System.out.print("\nVoulez-vous rejouer ? (o/n) : ");
        rejouer = scanner.next().toLowerCase().charAt(0) == 'o';
    } while (rejouer);

    System.out.println("Merci d'avoir joué !");
    scanner.close();
}
```

Conclusion

Ce TP m'a aidé à mieux comprendre la programmation Java. J'ai manipulé des tableaux, des structures de contrôle et découvert comment l'orienté objet rend le code plus clair. Ça m'a permis de transformer la théorie en pratique utile pour la suite.